



INVESTOR IN PEOPLE

The Patent Office  
Concept House  
Cardiff Road  
Newport  
South Wales  
NP10 8QQ



I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

Signed

*M. E. King*

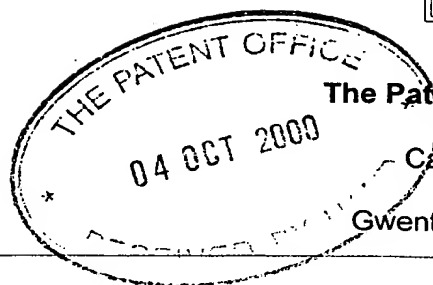
Dated

*24 September 2001*

**THIS PAGE BLANK (USPTO)**

# Request for a grant of a patent

(See the notes on the back of this form you can also get an explanatory leaflet from the Patent Office to help you fill in this form)



1. Your reference P009836GB

2. Patent application number  
(The Patent Office will fill in this part)

0024311.3

04 OCT 2000

0500T00 E573476-19 D02246  
\_P01/7700 0.00-0024311.3

3. Full name, address and postcode of the  
or of each applicant  
(underline all surnames)

ARM Limited  
110 Fulbourn Road  
Cherry Hinton  
Cambridge  
CB1 9NJ  
United Kingdom

7498124002

Patents ADP number (if you know it)

If the applicant is a corporate body, give  
the country/state of its incorporation

United Kingdom

4. Title of the invention

SINGLE INSTRUCTION MULTIPLE DATA  
PROCESSING

5. Name of your agent (if you have one)

D YOUNG & CO

"Address for service" in the United Kingdom  
to which all correspondence should be sent  
(including the postcode)

21 NEW FETTER LANE  
LONDON  
EC4A 1DA

Patents ADP number (if you know it)

59006

6. If you are declaring priority from  
one or more earlier patent  
applications, give the country and  
date of filing of the or each of these  
earlier applications and (if you know  
it) the or each application number

Country

Priority application  
number  
(if you know it)

Date of filing  
(day/month/year)

1st

2nd

3rd

7. If this application is divided or otherwise  
derived from an earlier UK application,  
give the number and filing date of the  
earlier application

Number of earlier  
application

Date of filing  
(day/month/year)

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if:  
a) any applicant named in part 3 is not an inventor, or  
b) there is an inventor who is not named as an applicant, or  
c) any named applicant is a corporate body.  
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	0
Description	7
Claim(s)	3
Abstract	1
Drawing(s)	4

10. If you are also filing any of the following, state how many against each item

Priority Documents	0
Translation of Priority Documents	0
Statement of inventorship and right to grant of a patent (Patents Form 7/77)	3
Request for preliminary examination and search (Patents Form 9/77)	1
Request for substantive examination (Patents Form 10/77)	0
Any other documents (Please specify)	0

11.

I/We request the grant of a Patent on the basis of this application.

Signature

Date

**D YOUNG & CO**  
Agents for the Applicants

4 Oct 2000

12. Name and daytime telephone number of person to contact in the United Kingdom

Nigel Robinson

023 80634816

### Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

### Notes

- If you need help to fill in this form or you have any questions, please contact the Patent Office on 01645 500505.
- Write your answers in capital letters using black ink or you may type them.
- If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheets should be attached to this form.
- If you answered 'Yes' Patents Form 7/77 will need to be filed.
- Once you have filled in the form you must remember to sign and date it.
- For details of the fee and ways to pay please contact the Patent Office.

**Statement of inventorship and  
of right to grant of a patent**



**The Patent Office**  
Cardiff Road  
Newport  
Gwent NP9 1RH

1. Your reference P009836GB
2. Patent application number (if you know it) **0024311.3**  
04 OCT 2000
3. Full name of the or of each applicant ARM Limited
4. Title of the invention SINGLE INSTRUCTION MULTIPLE DATA  
PROCESSING
5. State how the applicant(s) derived the right from the inventor(s) to be granted a patent By Virtue of Employment
6. How many, if any, additional Patents Forms 7/77 are attached to this form? (see note (c)) 0

7.

I/We believe that the person(s) named over the page (and on any extra copies of this forms) is/are the inventor(s) of the invention which the above patent relates to.

Signature

Date



**D YOUNG & CO**  
Agents for the Applicants

4 Oct 2000

8. Name and daytime telephone number of person to contact in the United Kingdom 023 80634816 Nigel Robinson

**Notes**

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.

b) Write answers in capital letters using black ink or you may type them.

c) If there are more than three inventor, please write the names and addresses of the other inventors on the back of another Patents Form 7/77 and attach it to this form.

d) When an application does not declare any priority, or declares priority from an earlier UK application, you must provide enough copies of this form so that the Patent Office can send one to each inventor who is not an applicant.

e) Once you have filled in the form you must remember to sign and date it.

D Young & Co ref: P009836GB

Enter the full names, addresses and postcodes of the inventors in the boxes and underline the surnames

Surname	<u>SYMES</u>
First Names	Dominic Hugo
Address	3 Applewood Close Cherry Hinton Cambridge CB1 9NU United Kingdom
Patents ADP number (if you know it): 7994601001	

Surname	<u>SEAL</u>
First Names	David James
Address	68 Keynes Road Cambridge CB5 8PR United Kingdom
Patents ADP number (if you know it): 6254312001	

Surname	
First Names	
Address	
Patents ADP number (if you know it):	

**Reminder:**  
Have you signed the form?

SINGLE INSTRUCTION MULTIPLE DATA PROCESSING

This instruction relates to the field of data processing systems. More particularly, this invention relates to data processing system in which it is desired to provide single instruction multiple data type operation.

5 Single instruction multiple data operation is a known technique whereby data words being manipulated in accordance with a single instruction in fact represent multiple data values within those data words with the manipulation specified being independently performed upon respective data values. This type of instruction can increase the efficiency with which a data processing system may operate and is  
10 particularly useful in reducing code size and speeding up processing operation. The technique is commonly, but not exclusively, applied to the field of manipulating data values representing physical signals, such as in digital signal processing applications.

When extending the data processing capabilities of a data processing system, an important consideration is the extent of any size, complexity, cost and power  
15 consumption overheads that may be introduced to support the additional processing capability. Measures that can add processing capability whilst reducing the additional overhead incurred are strongly advantageous.

Viewed from one aspect the present invention provides apparatus for data processing, said apparatus comprising a shifting circuit, an arithmetic circuit; and an  
20 instruction decoder responsive to an instruction to control said shifting circuit and said arithmetic circuit to perform an operation upon a data word  $R_n$  and a data word  $R_m$ , wherein said operation yields a value given by selecting a plurality of non-adjacent multibit portions of said data word  $R_m$  to form a plurality of multibit portions each of bit length A; optionally shifting said plurality of multibit portions by a common shift  
25 amount to shifted bit positions; promoting each of said plurality of multibit portions from said bit length of A to a bit length of B to form a plurality to promoted multibit portions, such that said promoted multibit portions may be abutted to form a promoted data word P; and performing a plurality of independent arithmetic operations using as input operands respective bit position portions of bit length B from both said promoted  
30 data word P and said data word  $R_n$  to form a result data word  $R_d$ .

The invention provides a new data processing instruction within a data processing system that may serve to unpack data values held within a data word and also perform a single-instruction-multiple-data type arithmetic operation upon the unpacked data values. The invention recognises that by unpacking non-adjacent data values within a data word may be implemented with considerably less additional overhead than conventional unpacking instructions which unpack adjacent data values. In particular, the need for additional data pathways that can diverge the bit positions of previously adjacent data values may be avoided. Instead, for example, already present masking and word shifting circuitry may be utilised. Furthermore, the simplification of the unpacking function allows the possibility for a single instructions to also provide an arithmetic operation upon the operands without introducing processing cycle constraint problems.

Whilst in its general form the invention may be applied to the selection of non-adjacent multibit portions of arbitrary length compared to their promoted length, particularly efficient and convenient implementations are ones in which the selected multibit portions are half the length of the promoted multibit portions with the promoted multibit portions being abutted within a promoted data word so as to have a length equal to the length of the input data words to the operation.

It will be appreciated that the promotion in lengths of the selected multibit portions may occur in different ways. Two particularly useful ways in which this may take place are by sign extension or by leading zero extension.

It will be appreciated that the arithmetic operation that is combined with the unpacking could take various different forms. However, particularly preferred embodiments are ones in which the arithmetic operation is an add operation performed independently upon respective promoted multibit portions. This instruction is particularly useful in many real life data processing situations, such as during the calculation of a sum of absolute differences as part of MPEG motion compensation calculations.

As previously mentioned, the invention is able to make use of existing processing resources within a data processing system in a more effective manner. This is particularly the case in a system in which a shifting circuit is provided upstream of an arithmetic circuit within the data path. Such an arrangement allows the unpacking,



with any of the optional shifts, to be performed prior to an arithmetic operation.

In preferred embodiments, in order to provide the desired functionality without imposing additional processing cycle time constraints, a promoting circuit that is responsible for promoting the selected multibit portions in length (e.g. either by sign extension or leading zero extension) is provided in parallel with a portion of the shifting circuit and the range of common shift amounts that is specified is restricted such that a first part of the shifting circuit can be used in combination with the promoting circuit to perform the required operations without extending the time data value takes to propagate beyond that already allowed for to provide propagation through the full shifting circuit in other operations.

Viewed from another aspect the invention provides a method of data processing, said method comprising the steps of decoding and executing an instruction that yields a result given by selecting a plurality of non-adjacent multibit portions of said data word  $R_m$  to form a plurality of multibit portions of bit length  $A$  optionally shifting said plurality of multibit portions by a common shift amount to shifted bit positions; promoting each of said plurality of multibit portions from said bit length of  $A$  to a bit length of  $B$  to form a plurality of promoted multibit portions, such that said multibit portions may be abutted to form a promoted data word  $P$ ; and performing a plurality of independent arithmetic operations using as input operands respective bit position portions of bit length  $B$  from both said promoted data word  $P$  and said data word  $R_n$  to form a result data word  $R_d$ .

The invention also provides a computer program product storing a computer program for controlling a general purpose computer in accordance with the above techniques, such as including a data processing instruction having the form of operation described above.

Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 schematically illustrates the action of a first SIMD type data processing instruction;

Figure 2 schematically illustrates a data path within a processing apparatus of a type well suited to executing the data processing instruction of Figure 1;

Figures 3 and 4 schematically illustrate two variants of a further SIMD type data processing instruction; and

Figure 5 schematically illustrates data path of a data processing system well suited for executing the data processing instructions of Figures 3 and 4.

5        Figure 1 illustrates the action of a first SIMD type data processing instruction termed ADD8TO16. This instruction comes in both signed and unsigned variants corresponding to the nature of the extension added to the front of a selected portion of each of the input operand data words as it is extended in length as part of the processing performed. The first input operand data word is stored within a register Rm  
10    of the data processing apparatus. The data word is formed of four 8-bit portions p0, p1, p2 and p3. Depending upon whether or not a rotate right operation of 8-bit positions is specified in the instruction, either the multibit portions p0 and p2 or alternatively the multibit portions p1 and p3 are selected out of the input data word within register Rm. The example illustrated in Figure 1 shows the non-adjacent  
15    portions p0 and p2 being selected in the unrotated (shifted) variant with the other variant being indicated by the dotted lines.

When the multibit portions have been selected, each is promoted in length from 8 bits to 16 bits using either zero or sign extension. The shaded portions of the promoted data word P shown in Figure indicate these extension portions.

20        The second input data word is stored within a register Rn and comprises two 16-bit data values. The example illustrated performs a single-instruction-multiple-data add operation whereby the extended p0 value is added to the lower 16 bit value a0 of Rn whilst the extended p2 value is added to the upper 16 bit portion a2 of the Rn value. This type of addition is one which may be considered as a full width addition  
25    with the carry chain broken between the 15<sup>th</sup> and 16<sup>th</sup> bits of the result. It will be appreciated that other SIMD type arithmetic operations may be performed, such as, for example, a SIMD subtraction.

The output result data word generated by the instruction of Figure 1 produces in the lower 16 bits the sum of p0 and a0 whilst the upper 16 bits contain the sum of p2  
30    and a2. This instruction is particularly useful in operations that determine the sum of absolute differences between respective data values whereby the a0 and a2 represent accumulate values with the values p0 to p3 representing individual absolute values of

signal difference values, such as pixel difference values. This type of operation is commonly needed in MPEG motion estimation processing and the ability to perform this operation at high speed is strongly advantageous.

Figure 2 illustrates an example data path 2 of a data processing system that may be used to implement the instruction of Figure 1. A register bank 4 holds 32-bit data words to be manipulated. Both the input operand data words stored in Rm and Rn are read from this register bank and the result data word is written back to register Rd in the register bank 4. The data path 2 includes a shifting circuit 6 and an adder circuit 8. The many other data processing instructions provided by the system utilise this shifting circuit 6 and adder circuit 8 in various different ways. Such a data path 2 is carefully designed so that the time taken for a data value to propagate through the shifting circuit 6 and the adder circuit 8 is well matched to the data processing cycle time. Efficient use of the hardware resources of the data path 2 is made in systems in which those resources are active for a high proportion of every data word propagating through the data path 2. A sign/zero extending and masking circuit 10 is provided in parallel with lower portion of the shifting circuit 6. A multiplex 12 is able to select either the output of the full shifting circuit 6 or the output of the sign/zero extending and masking circuit 10 as one of the inputs to the adder circuit 8. The other input to the adder circuit 8 is the input operand data word of Rn.

When executing the instruction of Figure 1, the input operand data word of Rm is supplied to the shifting circuit 6 in which an optional right shift of 8-bit positions is applied to the data word in dependence-upon whether or not that parameter was specified within the instruction. Within a multilevel multiplexer based shifter, such a restricted possibility shift may be provided relatively simply from a first portion of the shifting circuit 6 (e.g. in the case of a 32-bit system the first level of multiplexer may provide 16 bits of shift and the second level of multiplexer provides 8 bits of shift). Accordingly, a value optionally shifted by the specified amount can be tapped off from part way through the shifting circuit 6 and supplied to the sign/zero extending and masking circuit 10. This circuit 10 operates to mask out the non-selected multibit portions of the possibly shifted input operand data word of Rm and replace these masked out portions with either zeros or a sign extension of their respective selected multibit portions. The output of the sign/zero extending and masking circuit 10 passes

via a multiplexer 12 to a first input of the adder circuit 8. The second input of the adder circuit 8 is the input operand data word of  $R_n$ . The adder circuit 8 performs a SIMD add upon its inputs (i.e. two parallel 16-bit adds with the carry chain effectively broken between bit positions 15 and 16). The output of the adder circuit 8 is written  
5 back into register  $R_d$  of the a register bank 4.

Figures 3 and 4 illustrate two variants of a half word packing SIMD type instruction. The PKHTB instruction of Figure 3 takes a fixed top half of one input operand data word stored in register  $R_n$  and a variable position half bit portion of a second input operand data word stored in register  $R_m$  and combines these into  
10 respectively the top half and the bottom half of an output data word to be stored in register  $R_d$ . The instruction PKHBT takes the bottom half of an input operand data word of  $R_n$  and a variable position half word length portion of a second input operand data word of  $R_m$  and combines these respectively into the bottom and top halves of an output data word of  $R_d$ . It will be seen that the selected portion of the input operand  
15 data word of  $R_n$  in either case is unshifted in its location within the output data word  $R_d$ . This allows this portion to be provided by a simple masking or selecting circuit representing very little additional hardware overhead. The variable position half word portion of the instruction of Figure 3 is selected from bit positions 15 to 0 of the word of  $R_m$  after that word has been right shifted by  $k$  bit positions. Similarly, the half  
20 word length variable position portion of  $R_m$  selected in accordance with the instruction of Figure 4 is selected from bit positions 31 to 16 of the word of  $R_m$  after that word has been left shifted by  $k$  bit positions.

The variable shifting provided in combination with the packing function of the instructions of Figure 3 and Figure 4 is particularly useful for adjusting changes in the  
25 "Q" value of fixed point arithmetic values that can occur during manipulation of those values.

Figure 5 illustrates a data path 14 that is particularly well suited for performing the instructions of Figures 3 and 4. A register bank 16 again provides the input operand data words, being 32-bit data words in this example, and stores the output data  
30 word. The data path includes a shifting circuit 18, an adder circuit 20 and a selecting and combining circuit 22.

In operation, the unshifted input operand data word of Rn passes directly from the register bank 16 to the selecting and combining logic 22. In the case of instruction of Figure 3, the most significant 16 bits of the value of Rn are selected and form the corresponding bits within the output data word Rd. In the case of the instruction of Figure 4 it is the least significant 16 bits of the input operand data word of Rn that are selected and passed to form the least significant bits of the output data word Rd. The input operand data word of Rm passes through the full shifting circuit 18. In the case of the instruction of Figure 3, an arithmetic right shift of k bit positions is applied and then the least significant 16 bits from the output of the shifting circuit 18 are selected by the selecting and combining circuit 22 to form the least significant 16 bits of the output data word of Rd. In the case of the instruction of Figure 4, the shifting circuit 18 provides a left logical shift of k bit positions and supplies the result to the selecting and combining circuit 22. The selecting and combining circuit 22 selects the most significant 16 bits of the output of the shifting circuit 18 and uses these to form the most significant 16 bits of the output data word of Rd.

It will be seen that the selecting and combining circuit 22 is provided in a position in parallel with the adder circuit 20. Accordingly, given that the data path 14 is carefully designed to allow for a full shift and add operation to be performed within a processing cycle, the relatively straight forward operation of selecting and combining can be provided within the time period normally allowed for the operation of the adder circuit 20 without imposing any processing cycle constraints.

It will be understood that the data processing instructions explained above and as defined in the claims have been defined in terms of the result value achieved. It will be appreciated that the same result value can be achieved with many different processing steps and orders of steps. The invention encompasses all of these variants that produce the same final result value using a single instruction.

CLAIMS

1. Apparatus for data processing, said apparatus comprising:  
a shifting circuit;  
an arithmetic circuit; and  
5 an instruction decoder responsive to an instruction to control said shifting circuit and said arithmetic circuit to perform an operation upon a data word  $R_n$  and a data word  $R_m$ , wherein said operation yields a value given by:  
selecting a plurality of non-adjacent multibit portions of said data word  $R_m$  to  
form a plurality of multibit portions each of bit length  $A$ ;  
10 optionally shifting said plurality of multibit portions by a common shift amount to shifted bit positions;  
promoting each of said plurality of multibit portions from said bit length of  $A$  to a bit length of  $B$  to form a plurality of promoted multibit portions, such that said promoted multibit portions may be abutted to form a promoted data word  $P$ ; and  
15 performing a plurality of independent arithmetic operations using as input operands respective bit position portions of bit length  $B$  from both said promoted data word  $P$  and said data word  $R_n$  to form a result data word  $R_d$ .
2. Apparatus as claimed in claim 1, wherein  $B = 2 * A$ .  
20
3. Apparatus as claimed in any one of claims 1 and 2, wherein said plurality of multibit portions are shifted to shifted bit positions such that a lowest bit position multibit portion extends up from a zeroth order bit position.
- 25 4. Apparatus as claimed in any one of claims 1, 2 and 3, wherein promoting said multibit portions from a bit length of  $A$  to a bit length of  $B$  comprises one of:  
sign extending said multibit portions to a bit length of  $B$ ; and  
zero extending said multibit portions to a bit length of  $B$ .
- 30 5. Apparatus as claimed in any one of the preceding claims, wherein said plurality of independent arithmetic operations are independent add operations.

6. Apparatus as claimed in any one of the preceding claims, wherein said data words have a bit length of  $C$  and  $C = N * B$ , where  $N$  is an integer greater than 1.
- 5 7. Apparatus as claimed in claim 2 and 6, wherein  $C = B * 2$ .
8. Apparatus as claimed in any one of the preceding claims, wherein  $B = 16$  and  $A = 8$ .
- 10 9. Apparatus as claimed in any one of the preceding claims, wherein said common shift amount is  $B - A$ .
10. Apparatus as claimed in any one of the preceding claims, wherein said instruction is a single-instruction-multiple-data instruction.
- 15 11. Apparatus as claimed in any one of the preceding claims, wherein said instruction combines a data value unpack operation with an arithmetic operation.
12. Apparatus as claimed in any one of the preceding claims, wherein said shifting  
20 circuit is upstream of said arithmetic circuit in a data path of said apparatus.
13. Apparatus as claimed in any one of the preceding claims, wherein a promoting circuit operable to promote said multibit portions from a bit length of  $A$  to a bit length of  $B$  is disposed in parallel to a portion of said shifting circuit, said shifting circuit  
25 being operable to provide a restricted range of common shift amounts for data values passing through said shifting circuit when executing said instruction compared to a range of common shift amounts provided by said shifting circuit when executing other instructions.
- 30 14. A method of data processing, said method comprising the steps of decoding and executing an instruction that yields a value given by:

selecting a plurality of non-adjacent multibit portions of said data word  $R_m$  to form a plurality of multibit portions of bit length  $A$ ;

optionally shifting said plurality of multibit portions by a common shift amount to shifted bit positions;

5 promoting each of said plurality of multibit portions from said bit length of  $A$  to a bit length of  $B$  to form a plurality of promoted multibit portions, such that said promoted multibit portions may be abutted to form a promoted data word  $P$ ; and

performing a plurality of independent arithmetic operations using as input operands respective bit position portions of bit length  $B$  from both said promoted data  
10 word  $P$  and said data word  $R_n$  to form a result data word  $R_d$ .

15 15. A computer program product comprising a computer program for controlling a computer to perform a method as claimed in claim 14.

16. Apparatus substantially as hereinbefore described with reference to the  
15 accompanying drawings.

17. A method substantially as hereinbefore described with reference to the  
accompanying drawings.

20

18. A computer program product substantially as hereinbefore described with  
reference to the accompanying drawings.



**ABSTRACT****SINGLE INSTRUCTION MULTIPLE DATA PROCESSING**

A data processing system is provided with an instruction (ADD8TO16) that unpacks non-adjacent portions of a data word using sign or zero extension and  
5 combines this with a single-instruction-multiple-data type arithmetic operation, such as an add, performed in response to the same instruction. The instruction is well suited to use within systems having a data path (2) including a shifting circuit (6) upstream of an arithmetic circuit (8).

10 [Figure 1.]

**THIS PAGE BLANK (USPTO)**

1/4 .

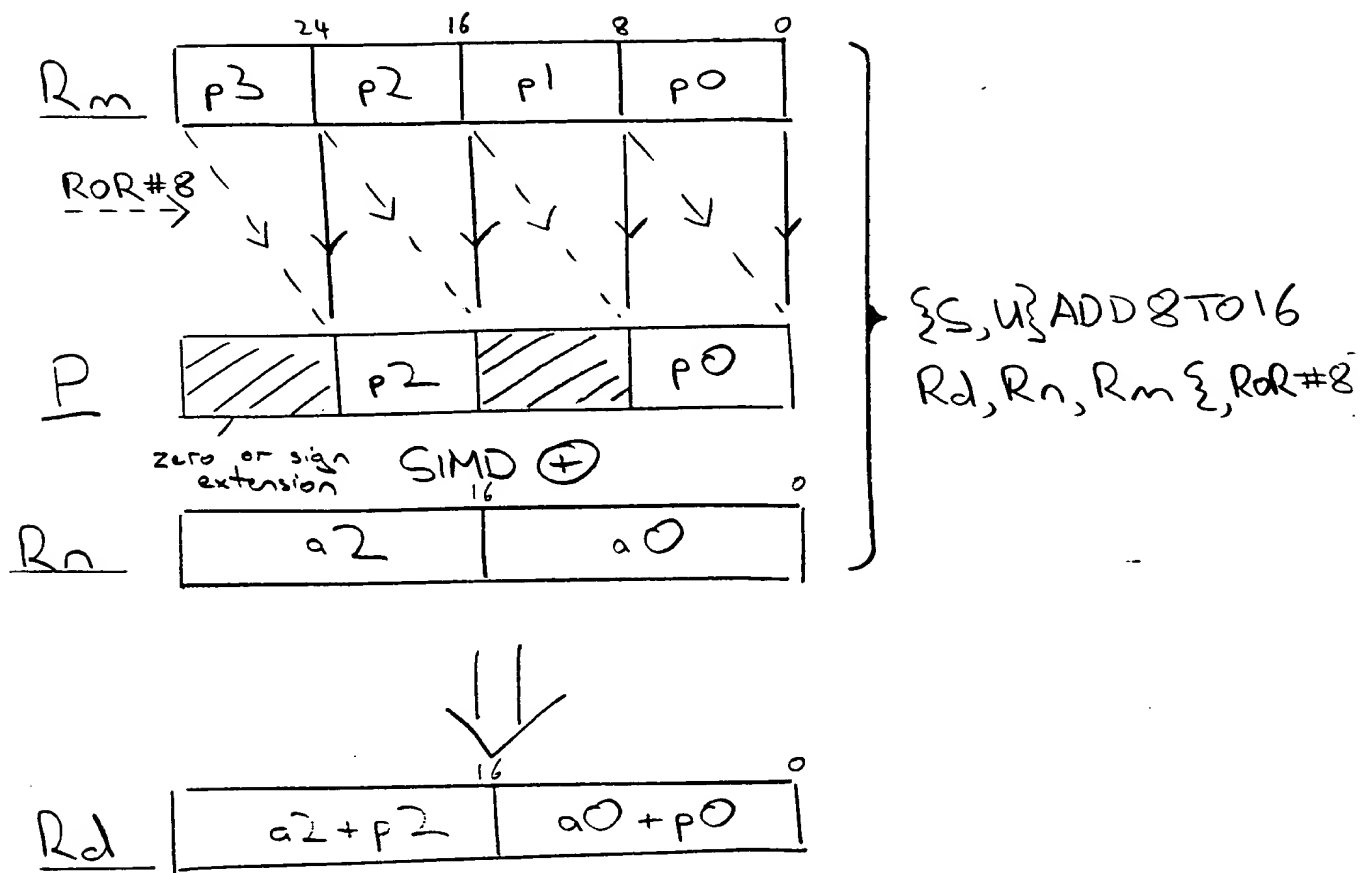


Fig. 1

**THIS PAGE BLANK (USPTO)**

2/4 .

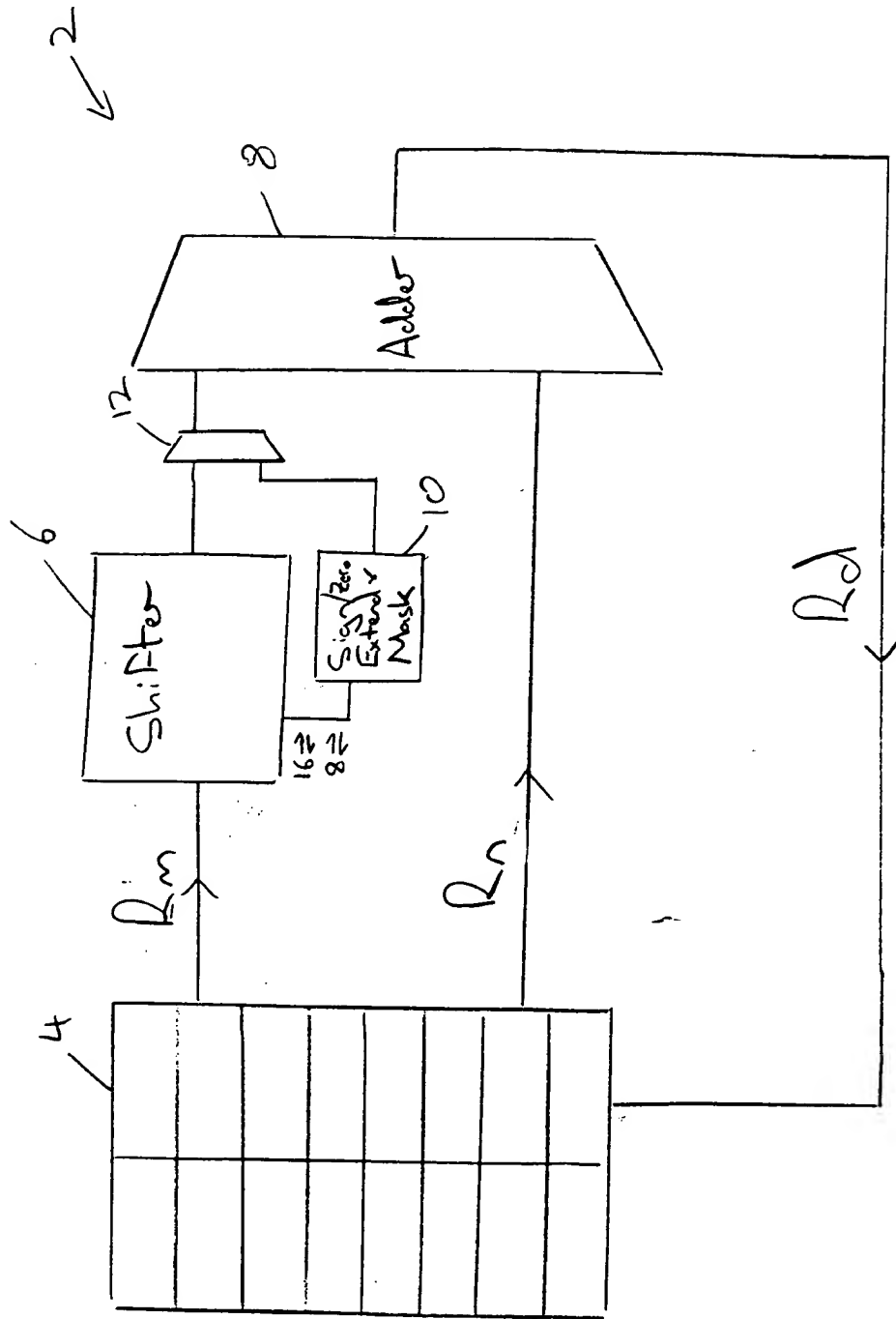


Fig. 2

**THIS PAGE BLANK (USPTO)**

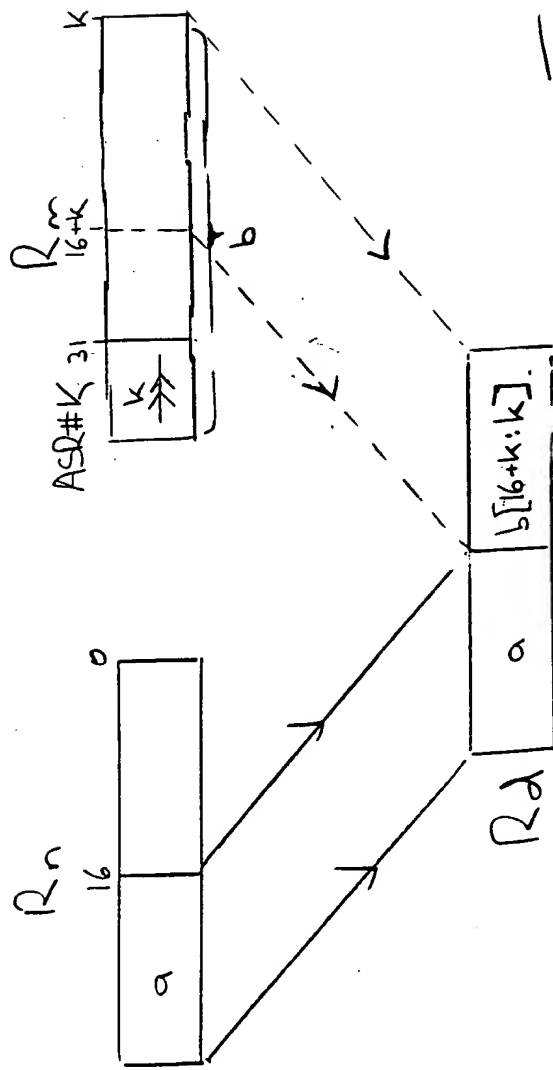


Fig. 3

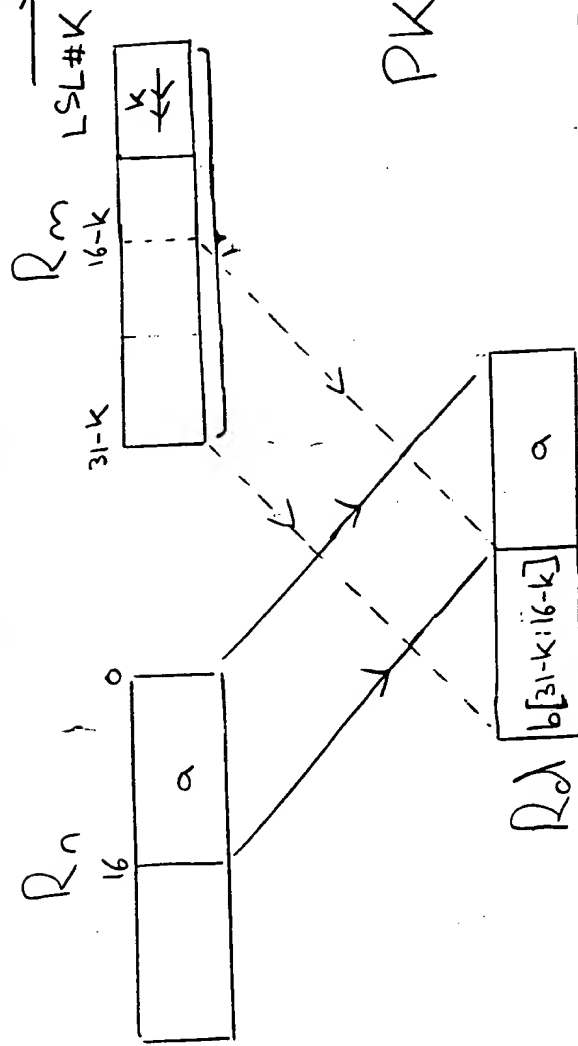


Fig. 4

PKH TB  $R_d, R_n, R_m, ASL\#k$

3/4.

PKH BT  $R_d, R_n, R_m, LSL\#k$

**THIS PAGE BLANK (USPTO)**



4/4

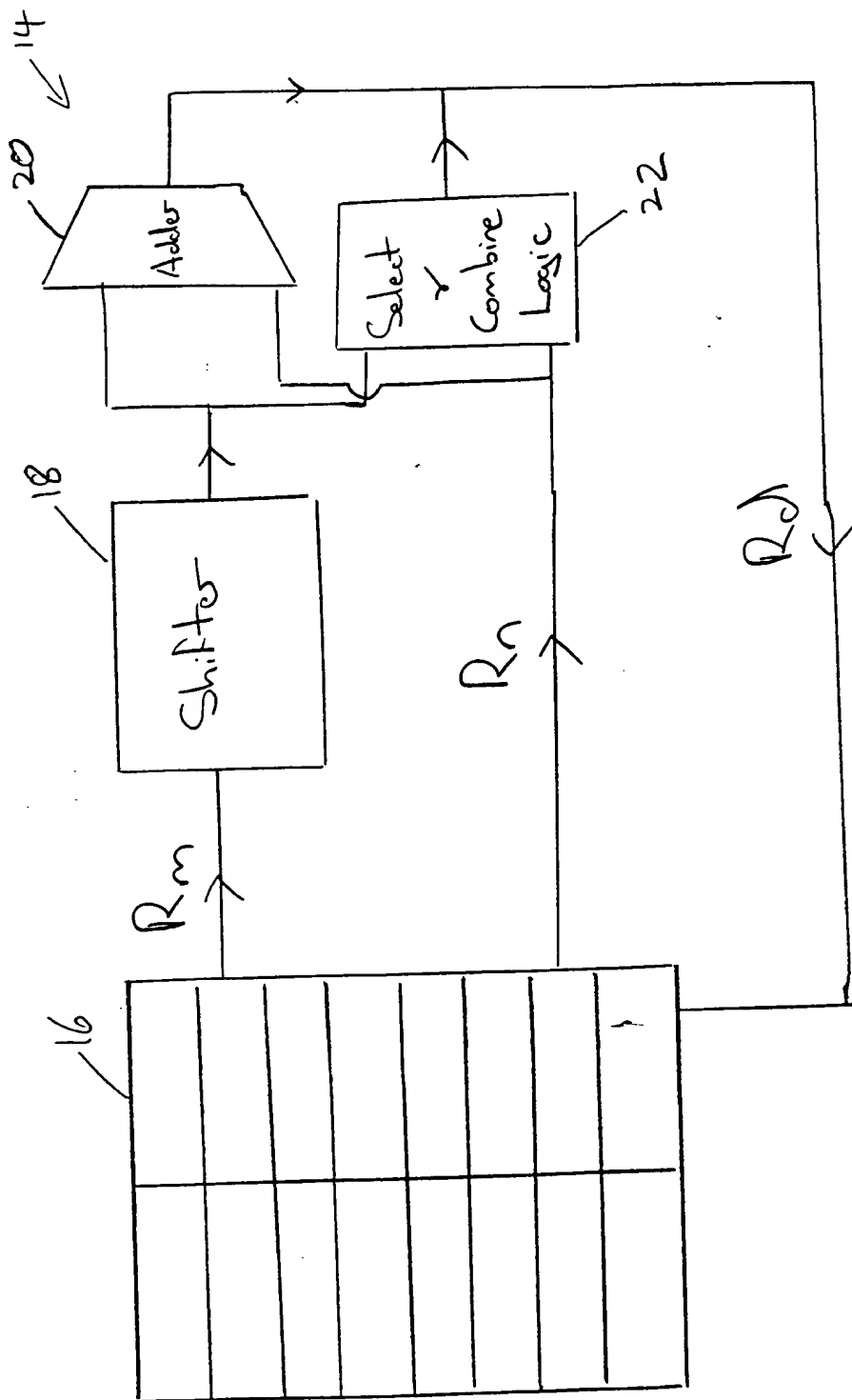


Fig. 5

09/957,467

**THIS PAGE BLANK (USPTO)**